Application No.: 10/705,946

IN THE CLAIMS:

1. (original)A programmable processor comprising: an instruction path; a data path; an external interface operable to receive data from an external source and communicate the received data over the data path; a cache operable to retain data communicated between the external interface and the data path; a register file operable to receive and store data from the data path and communicate the stored data to the data path; and an execution unit coupled to the instruction and data paths and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single instruction specifying both a shift amount and a register having a register width, the register containing a first plurality of data elements having an elemental width smaller than the register width, the number of data elements in the first plurality of data elements being inversely related to the elemental width, the shift amount configurable to an amount inclusively between zero and one less than the elemental width, the execution unit is operable to: (i) shift a subfield of each of the first plurality of data elements by the shift amount to produce a second plurality of data elements; and (ii) provide the second plurality of data elements as a catenated result.

2. (original)The processor of claim 1 wherein the catenated result is provided to a register.

3. (original)The processor of claim 1 wherein the shift amount is contained in a register specified by the instruction.

4. (original)The processor of claim 1 wherein the shift amount is contained in an immediate field of the instruction.

5. (original)The processor of claim 1 wherein each data element in the first plurality of data elements has a sign bit in a most significant bit position and wherein the execution unit is further operable to fill a shift amount number of most significant bits in each of the second plurality of data elements with the sign bit from a respective data element in the first plurality of data elements.

6. (original)The processor of claim 1 wherein the execution unit is further operable to fill a shift amount number of most significant bits in each of the second plurality of data elements with zeros.

7. (original)The processor of claim 1 wherein the execution unit is further operable to fill a shift amount number of bits in each of the second plurality of data elements with a subfield from a respective location of the contents of an additional register.

8. (original)The processor of claim 1 wherein the catenated result has a width of 128 bits.

9. (original)The processor of claim 1 wherein the elemental width of each of the first plurality of data elements is 32 bits.

10. (original) The processor of claim 1 wherein the elemental width of each of the first plurality of data elements is 16 bits.

11. (original)The processor of claim 1 wherein an elemental width of each of the second plurality of data elements is equal to the elemental width of each of the first plurality of data elements.

12. (original)The processor of claim 1 wherein an elemental width of each of the second plurality of data elements is equal to twice the elemental width of each of the first plurality of data elements.

13. (original)The processor of claim 1 wherein an elemental width of each of the second plurality of data elements is equal to half the elemental width of each of the first plurality of data elements.

14. (original)A method for shifting data in a programmable processor, the method comprising: decoding a single instruction specifying both a shift amount and a register having a register width, the register containing a first plurality of data elements having an elemental width smaller

than the register width, the number of data elements in the first plurality of data elements being inversely related to the elemental width, the shift amount configurable to an amount inclusively between zero and one less than the elemental width; shifting a subfield of each of the first plurality of data elements by the shift amount to produce a second plurality of data elements; and providing the second plurality of data elements as a catenated result.

15. (original)The method of claim 14 wherein the catenated result is provided to a register.

16. (original)The method of claim 14 wherein the shift amount is contained in a register specified by the instruction.

17. (original)The method of claim 14 wherein the shift amount is contained in an immediate field of the instruction.

18. (original)The method of claim 14 wherein each data element in the first plurality of data elements has a sign bit in a most significant bit position and wherein the method further comprises: filling a shift amount number of most significant bits in each of the second plurality of data elements with the sign bit from a respective data element in the first plurality of data elements.

19. (original)The method of claim 14 wherein the method further comprises: filling a shift amount number of bits in each of the second plurality of data elements with zeros.

20. (original)The method of claim 14 wherein the method further comprises: filling a shift amount number of bits in each of the second plurality of data elements with a subfield from a respective location of the contents of an additional register.

21. (original)The method of claim 14 wherein the catenated result has a width of 128 bits.

22. (original)The method of claim 14 wherein the elemental width of each of the first plurality of data elements is 32 bits.

23. (original)The method of claim 14 wherein the elemental width of each of the first plurality of data elements is 16 bits.

24. (original)The method of claim 14 wherein an elemental width of each of the second plurality of data elements is equal to the elemental width of each of the first plurality of data elements.

25. (original)The method of claim 14 wherein an elemental width of each of the second plurality of data elements is equal to twice the elemental width of each of the first plurality of data elements.

26. (New) A programmable processor comprising:

an instruction path;

a data path;

a plurality of registers operable to receive and store data from the data path and communicate the stored data to the data path; and

an execution unit coupled to the instruction path and the data path and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single group shift right instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register, the execution unit is operable to: (i) for each of the plurality of data elements in the operand register, shift a subfield of the data element towards the least significant bit by the shift amount and fill a shift amount number of most significant bits with the sign bit of the respective data element to produce a second plurality of equal-sized data elements and (ii) provide the second plurality of data elements as a catenated result to the destination register.

27. (New) The programmable processor set forth in claim 26 wherein the execution unit is operable to execute group shift right instructions on 16-bit, 32-bit and 64-bit signed integer data.

28. (New) The programmable processor set forth in claim 27 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

Application No.: 10/705,946

29. (New) The programmable processor set forth in claim 26 wherein the shift amount is contained in an immediate field of the instruction.

30. (New) The programmable processor set forth in claim 26 wherein the shift amount is contained in a register specified by the instruction.

31. (New) A programmable processor comprising:

an instruction path;

a data path;

a plurality of registers operable to receive and store data from the data path and communicate the stored data to the data path; and

an execution unit coupled to the instruction path and the data path and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single group shift right instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register, the execution unit is operable to: (i) for each of the plurality of data elements in the operand register, shift a subfield of the data element towards the least significant bit by the shift amount and fill a shift amount number of most significant bits with zeros to produce a second plurality of equal-sized data elements and (ii) provide the second plurality of data elements as a catenated result to the destination register.

32. (New) The programmable processor set forth in claim 31 wherein the execution unit is operable to execute group shift right instructions on 16-bit, 32-bit and 64-bit unsigned integer data.

33. (New) The programmable processor set forth in claim 32 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

34 (New) The programmable processor set forth in claim 31 wherein the shift amount is contained in an immediate field of the instruction.

35. (New) The programmable processor set forth in claim 31 wherein the shift amount is contained in a register specified by the instruction.

36. (New) A programmable processor comprising:

an instruction path;

a data path;

a plurality of registers operable to receive and store data from the data path and communicate the stored data to the data path; and

an execution unit coupled to the instruction path and the data path and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single group shift left instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register, the execution unit is operable to: (i) for each of the plurality of data elements in the operand register, shift a subfield of the data element towards the most significant bit by the shift amount and fill a shift amount number of least significant bits with the sign bit of the respective data element to produce a second plurality of equal-sized data elements and (ii) provide the second plurality of data elements as a catenated result to the destination register.

37. (New) The programmable processor set forth in claim 36 wherein the execution unit is operable to execute group shift left instructions on 16-bit, 32-bit and 64-bit signed integer data.

38. (New) The programmable processor set forth in claim 37 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

39. (New) The programmable processor set forth in claim 36 wherein the shift amount is contained in an immediate field of the instruction.

40. (New) The programmable processor set forth in claim 36 wherein the shift amount is contained in a register specified by the instruction.

41. (New) A method for shifting data in a programmable processor, the method comprising:

decoding a single group shift right instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register;

for each of the plurality of data elements in the operand register, shifting a subfield of the data element towards the least significant bit by the shift amount and fill a shift amount number of most significant bits with the sign bit of the respective data element to produce a second plurality of equal-sized data elements; and

providing the second plurality of data elements as a catenated result to the destination register.

42. (New) The method set forth in claim 41 wherein the execution unit is operable to execute group shift right instructions on 16-bit, 32-bit and 64-bit signed integer data.

43. (New) The method set forth in claim 42 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

44 (New) The method set forth in claim 41 wherein the shift amount is contained in an immediate field of the instruction.

45. (New) The method set forth in claim 41 wherein the shift amount is contained in a register specified by the instruction.

46. (New) A method for shifting data in a programmable processor, the method comprising:

decoding a single group shift right instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register;

for each of the plurality of data elements in the operand register, shifting a subfield of the data element towards the least significant bit by the shift amount and fill a shift amount number of most significant bits with zeros to produce a second plurality of equal-sized data elements; and

providing the second plurality of data elements as a catenated result to the destination register.

47. (New) The method set forth in claim 46 wherein the execution unit is operable to execute group shift right instructions on 16-bit, 32-bit and 64-bit unsigned integer data.

48. (New) The method set forth in claim 47 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

49. (New) The method set forth in claim 46 wherein the shift amount is contained in an immediate field of the instruction.

50. (New) The method set forth in claim 46 wherein the shift amount is contained in a register specified by the instruction.

51. (New) A method for shifting data in a programmable processor, the method comprising:

decoding a single group shift left instruction specifying a shift amount, an operand register containing a plurality of equal-sized data elements stored in partitioned fields of the operand register and a destination register;

for each of the plurality of data elements in the operand register, shifting a subfield of the data element towards the most significant bit by the shift amount and fill a shift amount number of least significant bits with the sign bit of the respective data element to produce a second plurality of equal-sized data elements; and

providing the second plurality of data elements as a catenated result to the destination register.

52. (New) The method set forth in claim 51 wherein the execution unit is operable to execute group shift left instructions on 16-bit, 32-bit and 64-bit signed integer data.

53. (New) The method set forth in claim 52 wherein the operand register is a 128-bit register and the destination register is a 128-bit register.

54 (New) The method set forth in claim 51 wherein the shift amount is contained in an immediate field of the instruction.

55. (New) The method set forth in claim 51 wherein the shift amount is contained in a register specified by the instruction.